

Compressed Learning



University of Technology, Sydney

Tianyi Zhou

Faculty of Engineering and Information Technology

University of Technology, Sydney

A thesis submitted for the degree of

Doctor of Philosophy

31 August 2013

To my loving parents
Yongxi Zhou and Jingping Yao

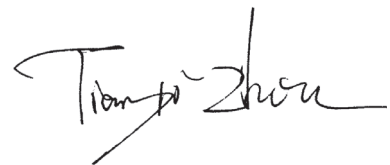
CERTIFICATE OF ORIGINAL AUTHORSHIP

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Student: Tianyi ZHOU

Date: 31/08/2013

A handwritten signature in black ink, appearing to read 'Tianyi Zhou', with a stylized, cursive script.

Acknowledgements

I benefited and learned a lot from my advisor, several professors, my friends, my colleagues and my family during PhD study in University of Technology, Sydney and Nanyang Technological University. I would like to take this good opportunity to appreciate their significant helps to me.

The first person I would like to express my appreciation and gratitude to is Professor Dacheng Tao. I am so lucky to have Prof. Tao as my professional mentor and academic advisor. I benefited significantly from various detailed discussions with him. Discussing a problem with him has been always a humbling but eye-opening experience, and he always gives me sufficient freedom to think and explore. His careful criticisms bridging both theory and application have greatly broadened my thought in research. His vision, creativeness and enthusiasm in solving challenging problems has greatly encouraged me and inspired my works. Without his high scientific criterion, endless patience, generous support, and constant guidance, this thesis cannot be accomplished. I also want to thank Prof. Tao as a nice friend, for his invaluable suggestions and experienced instructions on my research career and life.

I also wish to express my appreciation to other Professors who have gave me helpful guidance and encouragement during my PhD study and on conferences: Prof. Chengqi Zhang, Prof. Xindong Wu, Prof. Xingquan Zhu, Prof. Jieping Ye, Prof. Jerome H. Friedman, Prof. Trevor Hastie, Prof. Dean P. Foster and Prof. Emmanuel Candès. I learned a lot from discussions with them and their attitude for doing high-quality research.

I have been fortunate to work in a group gathering the most brilliant researchers and best friends in the past 5 years: Dr. Wei Bian, Dr. Jun Li, Dr. Bo Geng, Dr. Xinmei Tian, Dr. Zhang Zhang, Dr. Chao Zhang, Dr. Naiyang Guan, Dr. Lusong Li, Dr. Xiaoguang Rui, Dr. Weifeng Liu, Dr. Shengzheng Wang, Yang Mu, Bo Xie, Dongjing Song, Si Si, Yuanyuan Fu, Yong Luo, Yangxi Li, Zhibin Hong, Nannan Wang, Mingming Gong, Lianyang Ma, Maoying Qiao, Xiaoyan Li, Tongliang Liu, Fei Gao, Changxin Ding, Jie Gui, Xinchao Wang, Tianhao Zhang. Especially, I am deeply indebted to Dr Wei Bian, who gave me strong motivation and critical guidance when I have meet difficulties in research. He spent much time to teach me valuable things that I cannot easily learn by myself, especially at the beginning of my PhD study. I learned lots from discussion and collaboration with all group members in statistics, machine learning and optimization. Moreover, I enjoyed the invaluable friendships with them, their kindly support and accompany are always my source of strength and courage in both research and daily life. I own my deepest thanks to all of them!

I am also grateful to all the other friends who made my 5 years at Singapore and Sydney unforgettable: Guodong Long, Jing Jiang, Chunyang Liu, Meng Fang, Shirui Pan, Mingsong Mao, Hongshu Chen, Can Wang, Junfu Yin, Guoxin Su, Dianshuang Wu, Yi Ji, Ming Xie, Xiang Li, Yifan Li, Qian Sun, and my dearest friends Taoyu Lin and Peng Su since my college. They are the ones who have given me support during both joyful and stressful times, to whom I will always be thankful.

Finally, it is my greatest honor to thank my family: my parents, my grandparents, my uncle and auntie. They are always believing in me, keeping encouraging me, giving me indispensable suggestions, and fully supporting all my final decisions. No words could possibly express my deepest gratitude for their endless love, self-sacrifice and unwavering help. To them I dedicate this dissertation.

Abstract

There has been an explosion of data derived from the internet and other digital sources. These data are usually multi-dimensional, massive in volume, frequently incomplete, noisy, and complicated in structure. These “big data” bring new challenges to machine learning (ML), which has historically been designed for small volumes of clearly defined and structured data. In this thesis we propose new methods of “compressed learning”, which explore the components and procedures in ML methods that are compressible, in order to improve their robustness, scalability, adaptivity, and performance for big data analysis. We will study novel methodologies that compress different components throughout the learning process, propose more interpretable general compressible structures for big data, and develop effective strategies to leverage these compressible structures to produce highly scalable learning algorithms. We present several new insights into popular learning problems in the context of compressed learning. The theoretical analyses are tested on real data in order to demonstrate the efficacy and efficiency of the methodologies in real-world scenarios.

In particular, we propose “manifold elastic net (MEN)” and “double shrinking (DS)” as two fast frameworks extracting low-dimensional sparse features for dimension reduction and manifold learning. These methods compress the features on both their dimension and cardinality, and significantly improve their interpretation and performance in clustering and classification tasks.

We study how to derive fewer “anchor points” for representing large datasets in their entirety by proposing “divide-and-conquer anchoring”, in which the global solution is rapidly found for near-separable

non-negative matrix factorization and completion in a distributed manner. This method represents a compression of the big data itself, rather than features, and the extracted anchors define the structure of the data.

Two fast low-rank approximation methods, “bilateral random projections (BRP)” of fast computer closed-form and “greedy bilateral sketch (GreBske)”, are proposed based on random projection and greedy augmenting update rules. They can be broadly applied to learning procedures that requires updates of a low-rank matrix variable and result in significant acceleration in performance.

We study how to compress noisy data for learning by decomposing it into the sum mixture of low-rank part and sparse part. “GO decomposition (GoDec)” and the “greedy bilateral (GreB)” paradigm are proposed as two efficient approaches to this problem based on randomized and greedy strategies, respectively. Modifications of these two schemes result in novel models and extremely fast algorithms for matrix completion that aim to recover a low-rank matrix from a small number of its entries. In addition, we extend the GoDec problem in order to unmix more than two incoherent structures that are more complicated and expressive than low-rank or sparse matrices. The three proposed variants are not only novel and effective algorithms for motion segmentation in computer vision, multi-label learning, and scoring-function learning in recommendation systems, but also reveal new theoretical insights into these problems.

Finally, a compressed learning method termed “compressed labeling (CL) on distilled label sets (DL)” is proposed for solving the three core problems in multi-label learning, namely high-dimensional labels, label correlation modeling, and sample imbalance for each label. By compressing the labels and the number of classifiers in multi-label learning, CL can generate an effective and efficient training algorithm from any single-label classifier.

Contents

Contents	vii
List of Figures	xiii
List of Tables	xxii
Nomenclature	xxv
1 Introduction	1
1.1 Low-rank and Sparse Structures in Learning Problems	3
1.1.1 Low-rank Structure and Dimension Reduction	3
1.1.2 Sparse Structure and Sparse Learning	5
1.2 Literature Survey of Compressed Learning	7
1.3 Main Contributions and Road Map	10
2 Manifold Elastic Net: A Unified Framework for Sparse Dimension Reduction	16
2.1 Introduction	17
2.1.1 The proposed approach	18
2.2 Manifold Elastic Net	20
2.2.1 Part optimization	21
2.2.2 Whole alignment	22
2.2.3 Classification error minimization	24
2.2.4 Elastic net penalty	25
2.2.5 LARS for MEN	27
2.2.6 Fast LARS	32

2.2.7	Algorithm	33
2.2.8	Discussions	35
2.3	Experiments	38
2.4	Conclusion	50
3	Double Shrinking for Sparse Dimension Reduction	53
3.1	Introduction	54
3.1.1	Double shrinking model	55
3.1.2	Previous works	56
3.1.3	Main contribution	58
3.2	Definitions	59
3.2.1	Karush-Kuhn-Tucker conditions	59
3.2.2	Definitions	60
3.3	Double shrinking Algorithm	62
3.3.1	Initialization	62
3.3.2	Direction	63
3.3.3	Step size and update of A, B	68
3.3.4	Update of x, μ and η	70
3.3.5	Algorithm	70
3.3.6	Analyses and Proofs	72
3.4	Extensions of double shrinkage	75
3.4.1	Elastic net double shrinkage	75
3.4.2	Reweighted ℓ_1 double shrinkage	76
3.4.3	Structured double shrinkage	77
3.4.4	Sparse learning with multiple equality constraints	78
3.5	Relationships to existing techniques	79
3.5.1	Relationship to sparse PCA	80
3.5.2	Relationship to sparse coding	81
3.5.3	Relationship to LARS	84
3.6	Experiments	85
3.6.1	Classification	86
3.6.2	Nonlinear manifold learning	87
3.6.3	Data clustering	89

3.6.4	Feature selection	91
3.6.5	Scalability study	96
3.7	Conclusion	97
4	Divide-and-Conquer Anchoring for Near-separable Nonnegative Matrix Factorization and Completion	102
4.1	Introduction	103
4.1.1	Separable Nonnegative Matrix Factorization	104
4.1.2	Related Works	106
4.1.3	Motivation and Main Contributions	107
4.2	Divide-and-Conquer Anchoring (DCA)	109
4.2.1	Divide Step: Anchoring on Low-dimensional Projections	109
4.2.2	Conquer Step: Hypothesis Testing	112
4.2.3	DCA for Incomplete Data Matrix	113
4.2.4	Analysis: the Number of Sub-problems	116
4.3	Rapid Anchoring in 1D or 2D Space	117
4.3.1	Seeking Vertices of Convex Hull in 1D Space	118
4.3.2	Seeking Extreme Rays of Conical Hull in 2D Space	118
4.4	Numerical Results	120
4.4.1	Empirical Study on Synthetic Data	121
4.4.2	Collaborative Filtering by Finding Representative Users	123
4.4.3	Reconstruction of Images, Texts and Handwritten Digits	124
5	Randomized and Greedy Strategies for Bilateral Low-rank Approximation	126
5.1	Bilateral Random Projections (BRP)	126
5.1.1	Introduction	126
5.1.2	Bilateral random projections (BRP) based low-rank approximation	128
5.1.2.1	Low-rank approximation with closed form	128
5.1.2.2	Power scheme modification	128
5.1.3	Approximation error bounds	129
5.1.4	Proofs of error bounds	132

5.1.4.1	Proof of Theorem 8	132
5.1.4.2	Proof of Theorem 9	134
5.1.4.3	Proof of Theorem 10	135
5.1.4.4	Proof of Theorem 11	137
5.1.4.5	Proof of Theorem 12	138
5.1.5	Empirical Study	140
5.2	Greedy Bilateral Sketch (GreBske)	142
5.2.1	Low-rank Approximation	142
5.2.2	Greedy Bilateral Sketch	143
5.2.3	Empirical Study	146
6	GO Decomposition and Randomized Low-rank + Sparse Decomposition	148
6.1	Introduction	149
6.2	Go Decomposition (GoDec)	151
6.2.1	Naïve GoDec	151
6.2.2	Fast GoDec via BRP based approximation	152
6.3	Convergence of GoDec	152
6.4	Experiments	159
6.4.1	RPCA vs. GoDec	159
6.4.2	Background modeling	160
6.4.3	Shadow/Light removal	161
6.5	Conclusion	162
7	Greedy Bilateral Paradigm and Greedy Low-rank + Sparse Decomposition	163
7.1	Introduction	164
7.2	Background and Problem Formulation	165
7.3	Greedy Bilateral (GreB) Paradigm	166
7.4	Greedy Bilateral Smoothing	167
7.5	Analysis	168
7.6	Experiments on Video Data	171

8	Randomized and Greedy Algorithms for Matrix Completion	173
8.1	Introduction to Low-rank Matrix Completion	173
8.2	GoDec for matrix completion	175
8.2.1	Model and Algorithm	175
8.2.2	Matrix Completion Experiments of GoDec	176
8.3	Greedy Bilateral Completion (GreBcom)	177
8.3.1	Model and Algorithm	177
8.3.2	Greedy Bilateral Completion	177
8.3.3	Matrix Completion Experiments of GreBcom	179
9	Three GoDec Variants Unmixing General Incoherent Structures	182
9.1	Introduction	183
9.2	Main Contributions of This Chapter	186
9.3	Shifted Subspace Tracking (SST) for Motion Segmentation	188
9.3.1	The Problem of Motion Segmentation	188
9.3.2	SST model	191
9.3.3	SST Algorithm	193
9.3.3.1	Initialization	193
9.3.3.2	Update of τ	194
9.3.3.3	Update of L	195
9.3.3.4	Update of S	197
9.3.4	Motion Segmentation Experiments of SST	198
9.4	Multi-label Subspace Ensemble for Multi-label Learning	200
9.4.1	The Problem of Multi-label Learning	200
9.4.2	MSE model	203
9.4.3	MSE Algorithm	205
9.4.3.1	MSE training: randomized decomposition	205
9.4.3.2	MSE prediction: group sparsity	207
9.4.4	Multi-label Prediction Experiments of MSE	208
9.5	Linear Functional GoDec for Learning Recommendation System	212
9.5.1	LinGoDec Model and Algorithm	213
9.5.2	Empirical Study of LinGoDec	214

10 Compressed Labeling on Distilled Labelsets	216
10.1 Introduction	217
10.1.1 Three problems	218
10.1.2 Previous works	219
10.1.3 The proposed method	223
10.2 Compressed labeling (CL) via random projections	226
10.2.1 Random projection signs of label matrix	226
10.2.2 Improved sample balance of CL labels	227
10.2.3 Mutual independence of CL labels	233
10.2.4 Classification via support vector machines	234
10.3 Recovery algorithm on distilled labelsets (DLs)	234
10.3.1 Labelset distilling method (LDM)	235
10.3.2 Joint distribution of two random projection signs	236
10.3.3 KL divergence test for recovery	238
10.3.4 Recovery bound	242
10.4 Discussion	255
10.4.1 Contributions to multi-label learning	255
10.4.2 Relationship with compressed sensing	257
10.4.3 Relationship with error-correcting output codes	259
10.5 Experiments	263
10.5.1 Evaluation metrics	264
10.5.2 Datasets	265
10.5.3 Label compression and recovery	266
10.5.4 Multi-label prediction: comparison with BR	268
10.5.5 Multi-label prediction: comparison with other multi-label learning methods	283
10.5.6 Multi-label prediction: comparison with 2 SVM algorithms dealing with imbalanced data	287
10.5.7 Compression-performance trade-off	289
10.6 Conclusion	291
11 Conclusions	294

List of Figures

1.1	Relationships between the proposed approaches in this thesis. . .	11
2.1	Sample face images from the three databases. The first row comes from UMIST; the second row comes from FERET; and the third row comes from YALE.	39
2.2	Recognition Rate vs. Dimension on FERET	40
2.3	Recognition Rate vs. Dimension on UMIST	41
2.4	Recognition Rate vs. Dimension on YALE	42
2.5	Boxplot of Recognition Rate vs. Dimension (from 21 to 30) on FERET with 4 (5) training samples per person. For every dimension, from left to right, the seven boxes refer to MEN, DLA, LPP, NPE, FLDA, PCA, and SPCA.	44
2.6	Boxplot of Recognition Rate vs. Dimension (from 10 to 19) on UMIST with 5 (7) training samples per person. For every dimension, from left to right, the seven boxes refer to MEN, DLA, LPP, NPE, FLDA, PCA, and SPCA.	45
2.7	Boxplot of Recognition Rate vs. Dimension (from 5 to 14) on YALE with 5 (7) training samples per person. For every dimension, from left to right, the seven boxes refer to MEN, DLA, LPP, NPE, FLDA, PCA, and SPCA.	46
2.8	Plots of first 10 bases obtained from 7 dimensionality reduction algorithms on FERET For each column, from top to bottom: MEN, DLA, LPP, NPE, FLDA, PCA, and SPCA	47

LIST OF FIGURES

2.9	Plots of first 10 bases obtained from 7 dimensionality reduction algorithms on UMIST For each column, from top to bottom: MEN, DLA, LPP, NPE, FLDA, PCA, and SPCA	48
2.10	Plots of first 10 bases obtained from 7 dimensionality reduction algorithms on YALE For each column, from top to bottom: MEN, DLA, LPP, NPE, FLDA, PCA, and SPCA	49
2.11	Entries of one column of projection matrix vs. its ℓ_1 -norm in one LARS loop of MEN	50
2.12	Coefficient paths of 10 entries (features) in one column vector . .	51
3.1	(FERET) Recognition rate vs. Subspace dimensions curves of LDA, PCA, NPE and their double shrinkage versions on FERET face dataset. The first 10 dense eigenfaces obtained via eigenvalue decomposition (the top row) and the corresponding 10 66% sparse eigenfaces obtained via double shrinkage (the bottom row) are shown on the bottom of each plot.	88
3.2	(UMIST) Recognition rate vs. Subspace dimensions curves of LDA, PCA, NPE and their double shrinkage versions on UMIST face dataset. The first 10 dense eigenfaces obtained via eigenvalue decomposition (the top row) and the corresponding 10 66% sparse eigenfaces obtained via double shrinkage (the bottom row) are shown on the bottom of each plot.	89
3.3	(YALE) Recognition rate vs. Subspace dimensions curves of LDA, PCA, NPE and their double shrinkage versions on YALE face dataset. The first 10 dense eigenfaces obtained via eigenvalue decomposition (the top row) and the corresponding 10 66% sparse eigenfaces obtained via double shrinkage (the bottom row) are shown on the bottom of each plot.	90

3.4	(ORL) Recognition rate vs. Subspace dimensions curves of LDA, PCA, NPE and their double shrinkage versions on ORL face dataset. The first 10 dense eigenfaces obtained via eigenvalue decomposition (the top row) and the corresponding 10 66% sparse eigenfaces obtained via double shrinkage (the bottom row) are shown on the bottom of each plot.	91
3.5	(MNIST) Recognition rate vs. Subspace dimensions curves of LDA, PCA, NPE and their double shrinkage versions on MNIST handwritten digit dataset. The corresponding projection matrices are 66% sparse.	91
3.6	(USPS) Recognition rate vs. Subspace dimensions curves of LDA, PCA, NPE and their double shrinkage versions on USPS handwritten digit dataset. The corresponding projection matrices are 66% sparse.	92
3.7	(3D face) Two-dimensional embedding (with neighborhood graph of the original data) of 698 64×64 face images via double shrinking-ISOMAP. The images were sampled from a face rendered with different poses. Illumination differences were artificially eliminated. 50% of the face images have sparse representations in the two-dimensional subspace and thus are projected on the two coordinate axes X and Y . We sample 21 images from each axis and show them on the top and right of this figure, respectively.	93
3.8	(COIL-20) Two-dimensional embedding (with neighborhood graph of the original data) of 144 32×32 images of two objects (a toy cat and a toy duck) via double shrinkage-LLE. The images were sampled from a toy cat and a toy duck rendered with different poses. 90% of the images have sparse representations in the two-dimensional subspace and thus are projected on the two coordinate axes X and Y . We sample 21 images from each axis and show them on the top and right of this figure, respectively.	94

3.9	(Breast cancer) Sum of squares vs. Subspace dimensions (left), Accuracy vs. Subspace dimensions (middle), Normalized mutual information vs. Subspace dimensions (right) of clustering results on low dimensional representations of breast cancer data via PCA and double shrinkage-PCA. There are 60% samples owing zero representations on each coordinate obtained via double shrinkage.	95
3.10	(Wine) Sum of squares vs. Subspace dimensions (left), Accuracy vs. Subspace dimensions (middle), Normalized mutual information vs. Subspace dimensions (right) of clustering results on low dimensional representations of wine data via PCA and double shrinkage-PCA. There are 60% samples owing zero representations on each coordinate obtained via double shrinkage.	95
3.11	(Semeion) Sum of squares vs. Subspace dimensions (left), Accuracy vs. Subspace dimensions (middle), Normalized mutual information vs. Subspace dimensions (right) of clustering results on low dimensional representations of Semeion handwritten digit data via PCA and double shrinkage-PCA. There are 60% samples owing zero representations on each coordinate obtained via double shrinkage.	96
3.12	(Pitprops) Variance vs. Cardinality curves for the first 3 sparse principle components of the covariance matrix of Pitprops data obtained via double shrinkage and their corresponding solution paths. In the Variance vs. Cardinality plot, the red dash-dot line on the top of is the variance of the corresponding dense principle component, the red cross on each curve marks the corresponding selected principle component. In the solution path plot, the vertical red dash-dot line in each plot marks the step at which the sparse principle component is selected, curves with different colors represent the change of different variables.	98

3.13	Trade-off curves between explained variance and cardinality for the first sparse principal component of colon cancer data (left) and lymphoma data (right). Different Sparse PCA methods (Greedy search, Path SPCA, SPC, Double shrinkage) are compared with each other. SPC computes 10 sparse solutions of different cardinalities, while the other methods computes 500 solutions to build their solution paths. Their corresponding time costs are listed on the bottom of each plot.	99
3.14	Trade-off curves between explained variance and cardinality for the first sparse principal component of a 100×100 gaussian random matrix (left) and a 500×500 gaussian random matrix (right), each entry of the matrix is sampled from an independent standard gaussian distribution. Different Sparse PCA methods (Sparse PCA, Greedy search, Path SPCA, SPC, Double shrinkage) are compared with each other. SPC computes 10 sparse solutions of different cardinalities, while the other methods computes 100 (left) or 500 (right) solutions to build their solution paths. Their corresponding time costs are listed on the bottom of each plot.	100
4.1	Sub-problem in the divide step of DCA: finding the low-dimensional anchors $Y_{\bar{A}}$ on hyperplane \mathbb{P} when all data points X are contained in a <i>convex hull</i> of k anchors (vertices) X_A	109
4.2	Sub-problem in the divide step of DCA: finding the low-dimensional anchors $Y_{\bar{A}}$ on hyperplane \mathbb{P} when all data points X are contained in a <i>conical hull</i> of k anchors (extreme rays) X_A	111
4.3	Finding the anchors of full observable data points (a complete 300×500 matrix of rank 10) in a conical hull of anchors on 30 noise levels and 4 sub-problem amounts (only for DCA). Each point in the plots is obtained by averaging the results of 20 random trails on 20 different matrices. DCA invoking 2D rapid anchoring in Section 4.3 is compared to SPA [97] and XRAY [145].	119

4.4	Finding the anchors of full observable data points (a complete 50×100 matrix of rank 10 each row is normalized to have unit ℓ_1 norm) in a convex hull of anchors on 25 noise levels and 4 sub-problem amounts (only for DCA). Each point in the plots is obtained by averaging 5 random trails on 5 different matrices. DCA invoking 1D rapid anchoring in Section 4.3 is compared to LP based method Hottopixx [23].	120
4.5	Finding the anchors of data points with massive missing values (an incomplete 50×100 matrix each entry is observed with probability <i>sampling ratio</i>) in a conical hull of anchors via solving 125 sub-problems by DCA on 4 noise levels. The left two plots show the results when sampling ratio varies between $[0.01, 0.31]$ and the rank k is fixed to 10, while the two plots on the right show the results when rank k varies between $[5, 50]$ and the sampling ratio is fixed to 0.15. Each point in the plots is obtained by averaging 20 random trails on 20 different matrices. The divide step of DCA uses 2D rapid anchoring in Section 4.3.	123
5.1	low-rank matrix recovery via BRP: the recovery time for matrices of different size and different rank.	141
5.2	low-rank approximation via BRP: the relative approximation error for a 1000×1000 matrix with standard normal distributed entries on different rank.	142
5.3	low-rank image compression via BRP on FERET: BRP compresses 700 40×40 face images sampled from 100 individuals to a 700×1600 matrix with rank 60. Upper row: Original images. Middle row: images compressed by SVD (6.59s). Bottom row: images compressed by BRP (0.36s).	143
5.4	Low-rank approximation performed by Lanczos method (L-SVD), randomized SVD (R-SVD) and GreBske (G-SVD) on $10^4 \times 10^4$ matrix whose entries are sampled from i.i.d. normal distribution, p (K in G-SVD) is the power parameter.	146

6.1	Background modeling results of four 200-frame surveillance video sequences in $X = L + S$ mode. Top left: lobby in an office building (resolution 128×160 , learning time 39.75 seconds). Top right: shopping center (resolution 256×320 , learning time 203.72 seconds). Bottom left: Restaurant (resolution 120×160 , learning time 36.84 seconds). Bottom right: Hall of a business building (resolution 144×176 , learning time 47.38 seconds).	160
6.2	Shadow/light removal of face images from four individuals in Yale B database in $X = L + S$ mode. Each individual has 64 images with resolution 192×168 and needs 24 seconds learning time. . .	161
7.1	Phase diagram for GreBsmo on 500×500 matrices. Low-rank component is generated as $L = UV$, where entries of U and V are sampled from $\mathcal{N}(0, 1/n)$. Entries of sparse component S are sampled as 1 or -1 with probability $\rho/2$ and 0 with probability $1 - \rho$. On the 30×30 grid of sparsity-rank/ n plane, 20 trials are performed for each (ρ, r) pair. L is said to be successfully recovered if its rel. err. $\leq 10^{-2}$. The phase diagram shows the successful recovery rate for each (ρ, r) pair.	169
7.2	Background modeling of GreBsmo on three video sequences, top row: Hall, 144×176 pixels, 500 frames; middle row: ShoppingMall, 256×320 pixels, 253 frames; bottom row: Bootstrap, 120×160 pixels, 500 frames.	171
8.1	Phase diagram for GreBcom on 1000×1000 matrices. On the 20×20 grid of sampling ratio-rank/ n plane, 10 trials are performed for each (ρ, r) pair. A matrix is said to be successfully recovered if rel. err. $\leq 10^{-3}$. The phase diagram shows the successful recovery rate for each (ρ, r) pair.	179
9.1	Background modeling and object flow tracking results of a 50-frame surveillance video sequence from Hall dataset with resolution 144×176	197

9.2	Background modeling and object flow tracking results of a 50-frame surveillance video sequence from Shoppingmall dataset with resolution 256×320	199
9.3	Phase diagram (left) and corresponding CPU seconds (right) for LinGoDec on 750×750 matrices. Low-rank weight matrix W is of size 750×500 , and is generated by $W = UV$, where entries of U and V are sampled from $\mathcal{N}(0, 1/750)$ and $\mathcal{N}(0, 1/750)$, respectively. Features of items in Z is sampled from $\mathcal{N}(0, 1/750)$. Entries of sparse anomaly S are sampled as 1 or -1 with probability $\rho/2$ and 0 with probability $1 - \rho$. Noise G has entries sampled from $\mathcal{N}(0, 10^{-3})$. On the 50×30 grid of sparsity-rank/n plane, 10 trials are performed for each (ρ, r) pair. W is said to be successfully recovered if its rel. err. $\leq 10^{-2}$. The phase diagram shows the successful recovery rate for each (ρ, r) pair.	214
10.1	Compressed labeling on distilled labelsets. In the training stage, CL first compresses the original label matrix Y into Z , which is the sign matrix of random projections of Y on Gaussian random matrix A . Then binary classifiers (such as SVM) corresponding to the training set $\{X, Z\}$ are independently learned and stored in W . Meanwhile, the frequently appeared label subsets in Y are extracted by labelset distilling method (LDM) and stored in the distilled labelsets (DLs) D . In the prediction stage, CL first predicts the new labels z of a given sample x via the binary classifiers W . Given A and D , the DLs appearing in z are identified by a KL-divergence test based recovery algorithm and indexed by Ω . The final prediction y is the union of all the appeared DLs.	224
10.2	Random projections of x, y on two random vectors α and β , which are drawn uniformly from a k -dimensional hypersphere. The signs of random projections are marked as “+” for positive and “−” for negative in the figure. The hyperplanes $W1$ and $W2$ are perpendicular to x and y , respectively.	229

LIST OF FIGURES

10.3	Plot of $\delta/(1+\delta) - \gamma$ as a function of $\gamma \in [0, 1/2)$ on 5000 points between 0 and $1/2$	250
10.4	Sample balance, label independence and recovery error rate on 21 datasets (1). From top to bottom: Bibtex, Corel5k, Mediamill, IMDB, Enron.	269
10.5	Sample balance, label independence and recovery error rate on 21 datasets (2). From top to bottom: Genbase, Medical, Emotions, Scene, Slashdot.	270
10.6	Sample balance, label independence and recovery error rate on 21 datasets (3). From top to bottom: Yahoo-Arts, Yahoo-Business, Yahoo-Computers, Yahoo-Education, Yahoo-Entertainment. . . .	271
10.7	Sample balance, label independence and recovery error rate on 21 datasets (4). From top to bottom: Yahoo-Health, Yahoo-Recreation, Yahoo-Reference, Yahoo-Science, Yahoo-Social. . . .	272
10.8	Sample balance, label independence and recovery error rate on 21 datasets (5) on Yahoo-Society.	273
10.9	Trade-off between label compression and 5 prediction performance metrics, time costs on 5 datasets. From top to bottom: Bibtex, Corel5k, Mediamill, Enron and Medical.	290

List of Tables

2.1	Best recognition rate (%) on three databases. For MEN, DLA, LPP (SLPP), NPE, LDA (FLDA), PCA, SPCA (Sparse PCA), the numbers in the parentheses behind the recognition rates are the subspace dimensions. Numbers in the second column denote the number of training samples per individual.	42
3.1	Time complexity per iteration round of Sparse PCA, DSPCA, rSVD, SPC, Greedy SPCA and Double Shrinkage for calculating one sparse vector solution with s nonzero entries. We have $s_A, s_B \leq s \leq \min\{n, p\}$	82
3.2	Total cardinality and proportion of explained variance of the first 6 sparse principal components obtained via different methods from pitprops data. The results of Sparse PCA, rSVD and Greedy SPCA are calculated from the sparse loading vectors published in [273], [201] and [176], respectively.	97
3.3	Time cost (CPU seconds) of Sparse PCA, Path SPCA (faster version of DSPCA), Greedy SPCA, SPC and Double Shrinkage on two gene datasets (colon cancer, lymphoma) and two artificial datasets (a 100×100 and a 500×500 Gaussian random matrix). Note the time cost of SPC denotes the time for computing 10 sparse solutions rather than all the solutions on a solution path.	101

LIST OF TABLES

4.1	Normalized mean absolute error (NMAE), root mean square error (RMSE) and CPU seconds of DCA and matrix completion on MovieLens. $n/m/k$ of 3 datasets: 100k(943/1682/10), 1M(6040/3952/10), 10M(69878/10677/10). Result format: NMAE/RMSE/CPU seconds.	124
4.2	Reconstruction error and CPU seconds of SPA, XRAY and DCA on three datasets. The rank k for reconstructing them is 30, 50, 50. Result format: ℓ_2 error/CPU seconds.	125
6.1	Relative error and time cost of RPCA and GoDec in low-rank+sparse decomposition tasks. The results separated by “/” are RPCA and GoDec, respectively.	159
7.1	Comparison of time costs in CPU seconds of PCP, GoDec and GreBsmo in low-rank and sparse matrix decomposition task on background modeling datasets.	172
8.1	Relative error and time cost of OptSpace and GoDec in matrix completion tasks. The results separated by “/” are SVT [35] (a nuclear norm minimization method), OptSpace [137] (a subspace optimization method on Grassmann manifold) and GoDec, respectively. See [137] for the results of the other methods, e.g., FPCA and ADMIRA.	176
8.2	Relative error and time cost of OptSpace, SVP, ADMiRA and GreBcom in matrix completion tasks of different matrix size and rank. Notations: $m(n)$ -square matrix size, r -rank, ρ -sampling ratio $ \Omega _0/mn$, rel. err.-relative error, time-CPU time, “-”-does not apply due to speed or divergence.	180

8.3	$RMSE_{test}$ /CPU time of OptSpace, SVP and GreBcom in matrix completion tasks on recommendation system data with different training set ratio (for MovieLens) or different number of test ratings per user (for Jester), “-”-does not apply due to speed or divergence. Size and rank information (m/n/r) of datasets: 100k(943/1682/3), 1M(6040/3952/10), 10M(69878/10677/10), J1(24983/100/10), J2(23500/100/10), J3(24938/100/10).	181
9.1	Information of datasets that are used in experiments of MSE. In the table, n (training samples+test samples) is the number of samples, p is the number of features, k is the number of labels, “Card” is the average cardinality of all label vectors.	209
9.2	Prediction performances (%) and CPU seconds of BR [216], ML-KNN [246], MDDM [253] and MSE on Yahoo. Prec-precision, Rec-recall, F1-F1 score, Acc-accuracy	210
9.3	Prediction performances (%) and CPU seconds of BR [216], ML-KNN [246], MDDM [253] and MSE on 8 datasets. Prec-precision, Rec-recall, F1-F1 score, Acc-accuracy	211
10.1	Information of datasets that are used in label compression and recovery experiments and multi-label prediction experiments. In the table, n refers to the number of samples, p refers to the number of features, k refers to the number of labels, K refers to the number of unique label vectors, “Card” refers to the average cardinality of all label vectors, “Density” refers to the average nonzero entry proposition of all label vectors.	266
10.2	Training set size, test set size and the obtained distilled labelsets size of each datasets in the multi-label prediction experiments. In order to compare the number of distilled labelsets d with the number of labels k and the number of unique labelsets K , we list k and K of each datasets in the table as well.	274

LIST OF TABLES

10.3	Multi-label performances and time costs of BR and CL on 21 datasets with different C parameters (1). HL-Hamming Loss, Prec-Precision, Rec-Recall, Acc-Accuracy, Time-CPU seconds, labels-Number of Labels in training stage. “-” denotes the failed experiment that incorrectly predicts all the test samples as negative. The best performances of BR and CL are highlighted with different colors.	275
10.4	Multi-label performances and time costs of BR and CL on 21 datasets with different C parameters (2).	276
10.5	Multi-label performances and time costs of BR and CL on 21 datasets with different C parameters (3).	277
10.6	Multi-label performances and time costs of BR and CL on 21 datasets with different C parameters (4).	278
10.7	Multi-label performances and time costs of BR and CL on 21 datasets with different C parameters (5).	279
10.8	Multi-label performances and time costs of BR and CL on 21 datasets with different C parameters (6).	280
10.9	Prediction performances and time costs of ML-knn, MDDM, ML-CS and CL on 10 datasets. “-” denotes the failed experiment whose time cost exceeds 10^5 secondes.	284
10.10	Prediction performances and time costs of ML-knn, MDDM, ML-CS and CL on 11 sub datasets from Yahoo dataset.	285
10.11	Prediction performances and time costs of SVM-SMOTE, SVM-WEIGHT and CL on 5 datasets.	288